***Задача 1***

#include <stdio.h>

int A( int a){

if(a>0) a\*=2;

else

if (a<0) a-=3;

else a=10;

printf("\nValue of \"a\": %d",a);

return a;

}

int main(int argc, char \*argv[])

{

int a;

printf("Enter \"a\": ");

scanf("%d",&a);

a= A(a);

return 0;

}

***Задача 2***

#include <stdio.h>

int convert(int a){

int b=0;

int c=1;

while(a!=0)

{

b+=(a%10)\*c; /\*цифру наименьшего разряда введенного числа

c\*=2; умножаем на степень числа 2, где показатель

степени равен номеру разряда цифры;

суммируем полученное значение степени с предыдущими вычислениями (если они производились) степеней меньших показателей \*/

a=(a-(a%10))/10; /\*находим целую часть числа от деления на 10, т.е.

избавляемся от младшего разряда\*/

}

printf("\nDecimal number: %d", b);

return a;

}

int main(int argc, char \*argv[])

{

int a;

printf ("Enter a binary number: ");

scanf ("%d", &a); // ввод двоичного числа (по сути это десятичное число )

a=convert(a);

return 0;

}

***Задача 3***

#include<stdio.h>

int power(int x, int y){

int i, n;

i=1;

n=x;

if (y==0) x=1; // проверка на нулевую степень

else {

while (y!=i) {

x\*=n;

i++;

}

}

return x;

}

int main (int argc, char \*argv[]){

int x, y; // x – основание степени, y – ее показатель

printf("Enter the base and the power exponent\n");

scanf("%d%d", &x, &y);

x=power(x, y);

printf("\nValue of power: %d", x);

return 0;

}

***Задача 4***

#include <stdio.h>

#include <math.h>

float square(float a, float b, float c){

float x1, x2, D;

if (a==0) // проверка на обращение в линейное уравнение

{

x1= -c/b;

printf("\nGiven equation turns into linear, x=%f", x1);

}

else {

D=b\*b-4\*a\*c; // находим дискриминант и кол-во корней, вычисляем их

if (D>=0) {

x1=(-b+(sqrtf(D)))/(2\*a);

x2=(-b-(sqrtf(D)))/(2\*a);

if (x1==x2) printf("Equation root: %f",x1);

else printf("\nEquation roots:\n%f\n%f", x1, x2);

}

else printf("\nNo solution");

}

}

int main(int argc, char \*argv[]){

float x, a, b, c;

printf("Enter coefficients of the equation\n");

scanf("%f%f%f", &a,&b,&c); // ввод коэффициентов уравнения

x= square(a,b,c);

return 0;

}

***Задача 5***

#include <stdio.h>

#include <stdlib.h>

#define N 5

int random(int x){

srand(time(NULL));

x = 1 + rand() % 10; // случайное число от 1 до 10

printf("Guess the number from 1 to 10. You have %d attempts.\n", N);

int i, t;

for (i=1; i<=N; i++) {

printf("Enter the number: ");

scanf("%d", &t);

if (t == x) {

printf("You've guessed!.");

break;

}

else printf("Wrong =( %d attempts left\n", N-i);

}

return x;

}

int main(int argc, char \*argv[])

{

int x;

x=random(x);

return 0;

}

***Задача 6***

#include <stdio.h>

float degree(float grad, int N, int i){

float cel, far, kel;

switch (N){

case 1:{ if(grad >= -273){

far=grad\*1.8+32;

kel=grad+273;

printf("Temperature in Fahrenheit: %f\nTemperature in Kelvin: %f\n", far, kel);

}

else printf("Entered temperature < absolute zero");

break;

}

case 2: { if(grad >= -459){

cel=(grad-32)/1.8;

kel=cel+273;

printf("Temperature in Celsius: %f\nTemperature in Kelvin: %f\n", cel, kel);

}

else printf("Error. Entered temperature < absolute zero");

break;

}

case 3: { if(grad >= 0){

cel=grad-273;

far=cel\*1.8+32;

printf("Temperature in Celsius: %f\nTemperature in Fahrenheit: %f\n", cel, far);

}

else printf("Entered temperature < absolute zero");

break;

}

}

return grad;

}

int main(int argc, char \*argv[]) {

int i;

int N=0;

float grad;

do {

printf("Choose the scale: 1- Celsius, 2 - Fahrenheit, 3 - Kelvin\n");

scanf("%d", &N); // ввод шкалы измерения

if ( N!=1 && N!=2 && N!=3 ) printf("Error. Try again.\n");

else {

printf("Enter temperature in chosen scale : ");

scanf("%f", &grad); // ввод численного значения температуры

grad= degree(grad, N, i);

}

printf("\n Enter '1' if you want to continue or '0' if you want to stop the program "); // возможность переводить сразу несколько значений

scanf("%d", &i);

printf("\n\n");

}while ( i != 0);

return 0;

}

***Задача 7***

#include <stdio.h>

int sredn(int N){

int i;

int k=0;

float sred;

float b=0;

do {

scanf("%d", &N); // ввод числа элементов массива

if (N<=0 || N>500) printf("\nError. Try again.\n");

} while (N<=0 || N>500);

float A [N];

printf("\nEnter the array elements\n");

// ввод элементов массива и подсчет положительных элементов

for (i=0; i<N; i++) {

scanf ("%f", & A[i]);

if (A[i]>0){

k++;

b+=A[i];

}

}

if (k==0) printf("\nNo positive elements");

else{

sred = b/k; /\* вычисление среднего арифметического положительных

элементов массива\*/

printf("The number of positive elements: %d\nArithmetic mean of positive elements: %f", k,sred);

}

return sred;

}

int main(int argc, char \*argv[]) {

int N;

printf(" Enter 0 < N =< 500 - the number of array elements\n");

N = sredn(N);

return 0;

}

***Задача 8***

#include <stdio.h>

int matrix(int N,int M,int K,int S){

// проверка возможности перемножения матриц

if (M != K) printf(" Matrices can't be multiplied.");

else {

float A [N][M];

float B [M][S];

float C [N][S];

int i,j,k;

// ввод элементов первой матрицы

printf("\n Enter 1st matrix's elements\n");

for (i=0; i<N; i++)

for (j=0; j<M; j++){

printf(" A[%d][%d]= ", i+1, j+1);

scanf ("%f", &A[i][j]);

}

// ввод элементов второй матрицы

printf("\n Enter 2nd matrix's elements\n");

for (j=0; j<M; j++)

for (k=0; k<S; k++){

printf(" B[%d][%d]= ", j+1, k+1);

scanf ("%f", &B[j][k]);

}

// перемножение двух матриц и вывод результирующей матрицы

float p;

for (i=0; i<N; i++)

for (k=0; k<S; k++){

p=0;

for (j=0; j<M; j++)

p+= A[i][j]\*B[j][k];

C[i][k]=p;

}

printf("\n\n");

for (i=0; i<M; i++)

for (k=0; k<S; k++)

printf(" C[%d][%d]= %f\n", i+1, k+1, C[i][k]);

}

}

int main(int argc, char \*argv[]) {

int N,M,K,S;

int x;

// задание размеров двух произвольных матриц

printf(" Enter size of the 1st matrix (NxM) : ");

scanf("%d %d", &N,&M);

printf(" Enter size of the 2nd matrix (KxS) : ");

scanf("%d %d", &K,&S);

x = matrix(N,M,K,S);

return 0;

}

***Задача 9***

#include <stdio.h>

int hex(char ch, int i, int k){

printf("Enter the string, to stop enter \".\"\n"); /\* Ввод строки символов (окончание ввода символ “.” -точка)\*/

do {

ch = getchar();

/\* Проверяем, является ли данный символ исходной строки шестнадцатеричным; вычисляем k и i \*/

if((ch >= '0' && ch <= '9') || (ch >= 'a' && ch <= 'f') ||(ch >= 'A' && ch <= 'F')) i++ ;

k++;

} while (ch != '.');

if ((k-2)==i) printf("This string IS a hex number");

else printf("This string IS NOT a hex number");

}

int main(int argc, char \*argv[]){

char ch, p;

int i, k, n; /\* k - кол-во всех введенных символов,

k=1; i - кол-во шестнадцатеричных символов \*/

do{

i= 0;

n=hex(ch, i, k);

printf("\n Enter '1' if you want to continue or '0' if you want to stop the program "); // возможность проверить несколько строк подряд

scanf("%d", &p);

printf("\n");

k=0;

}

while ( p != 0);

return 0;

}

***Задача 10***

#include <stdio.h>

#include <stdlib.h>

int sequence(int N){

int k=0;

int i;

int \*a;

a=(int\*) malloc(N\*sizeof(int)); //выделение памяти

if (a == NULL)

{

printf ("Memory allocation error.");

return 0;

}

printf("Enter the array\n"); // ввод массива и проверка на возрастание

scanf("%d", &a[0]);

for (i = 1; i < N; i++){

scanf("%d", &a[i]);

if (a[i] <= a[i-1]) k++;

}

if (k>0) printf("Array is not an increasing sequence.");

else printf("Array is an increasing sequence.");

}

int main(int argc, char \*argv[])

{

int N, x ;

printf("Enter the size of the array.\n"); //задаем размер массива

do{

scanf("%d", &N);

if (N <= 0) printf("Error. Try again.\n");

} while (N <= 0);

x=sequence(N);

return 0;

}

***Задача 11***

#include <stdio.h>

#include <stdlib.h>

int max\_row(int N, int M){

int i, j, k;

float S, MaxSum;

float \*\*a ;

k=0;

S=0;

MaxSum=0;

a=(float \*\*) malloc(N\*sizeof(float\*)); // выделение памяти

if (a == NULL)

{

printf ("Memory allocation error.");

return 0;

}

for (i=0; i<N; i++){

a[i]=(float \*) malloc(M\*sizeof(float));

}

printf("Enter the matrix elements.\n"); // ввод элементов матрицы

for (i=0; i<N; i++){

S=0;

for(j=0; j<M; j++) { // последовательно вычисляются суммы строк

scanf("%f", &a[i][j]);

S+=a[i][j];

}

if (S > MaxSum) { // находим номер строки с максимальной суммой

MaxSum=S;

k=i;

}

}

printf("The number of the row with the largest sum of elements: %d", k+1);

}

int main(int argc, char \*argv[]){

int N, M, x;

printf("Enter the size of the matrix.\n"); // задаем размер матрицы

do{

scanf("%d" "%d", &N, &M);

if (N<=0 || M<=0) printf("Error. Try again.\n");

} while (N<=0 || M<=0);

x = max\_row(N, M);

return 0;

}

***Задача 12***

#include <stdio.h>

#include <string.h>

#define MaxName 100

int check(int N){

int i, k;

k = 0;

char name[MaxName];

char a[N][MaxName]; /\* задаем фамилии

студентов\*/

printf("\n Enter the student's surnames\n");

gets(a[0]);

for (i = 0; i < N; i++){

printf(" Student %d: ", i+1);

gets(a[i]);

}

printf("\n Surname of a student you want to find: "); /\* вводим фамилию

интересующего нас студента \*/

gets(name);

for (i = 0; i < N; i++){

if (strcmp(a[i], name) == 0){ /\* последовательно сравниваем

введенную фамилию с элементами массива\*/

k=1;

break;

}

}

if (k == 1) printf("This student is in the group");

else printf("This student is NOT in the group");

}

int main (int argc, char \*argv[])

{

int N, x;

printf(" Enter the number of students: "); /\* задаем кол-во студентов

в группе \*/

scanf("%d", &N);

x=check(N);

return 0;

}